Experimental Demonstration of Policy-based Dynamic End-to-End Provisioning over Multi-Layer Network using SDN

J. Marhuenda, A. Aguado, S. Bidkar, E. Hugues-Salas, R. Nejabati and D. Simeonidou

High Performance Networks group, University of Bristol, UK. jaume.marhuenda@bristol.ac.uk

Abstract Contemporary transport networks require dynamic multi-layer service provisioning to support ever-growing application traffic. We present a software platform architecture based on SDN principles combined with a network policy engine to experimentally demonstrate dynamic multi-layer provisioning over a national dark-fibre network.

Introduction

The proliferation of mobile devices and the advent of bandwidth intensive applications served over cloud infrastructures are introducing significant traffic dynamics in access, metro and core segments of infrastructure networks. Over-the-top application providers and enterprise customers are leasing cloud infrastructure from IaaS providers for business agility and operational simplicity. With increasing adoption of cloud services, co-ordinated multi-layer network provisioning is essential to accommodate dynamic traffic demands while efficiently utilizing network resources. Recent advances in packet-optical integration and software defined networking (SDN) facilitate the centralized control and dynamic configuration of network resources. However, depending on the network architecture, multiple stake-holders and legacy systems, there is a limited network visibility and control for a single SDN controller to setup end-to-end service paths in a multi-domain, multi-technology network. In this context, it becomes challenging to represent all network elements using a common control abstraction that could express all functionalities for all underlying technologies. The OpenFlow abstraction initially proposed for realizing SDN systems for packet networks is not optimal for optical transport networks. There are a significant number of existing standards for transport network provisioning protocols such as GMPLS and PCE. However, they lack co-ordination with packet networks to achieve joint optimization of multi-layer networks. Therefore, in this paper we propose a software platform architecture combining a cloud management system, SDN controller for packet networks (virtual as well as physical) and an optical network provisioning system. This enables dynamic circuit setup in the optical domain to achieve cloud service provisioning over multi-layer networks. There are existing proposals in the literature and in standardization bodies for orchestration of cloud and multi-layer network resources based on static provisioning [1]. However, we propose a policy engine component in our architecture that provides a simplified interface for the network operators and virtual infrastructure tenants to specify their requirements and constraints. The policy-engine converts high-level user policies into a forwarding rules database. Such a database acts as a point of reference for multiple provisioning systems to setup reactive end-to-end paths while optimizing network resources. In this paper, we demonstrate the proposed software platform and the experimental evaluation over an operational metro region data centre interconnection network emulating multiple scenarios. In the following sections, we describe in detail the architecture of the proposed software platform as well as the workflow for

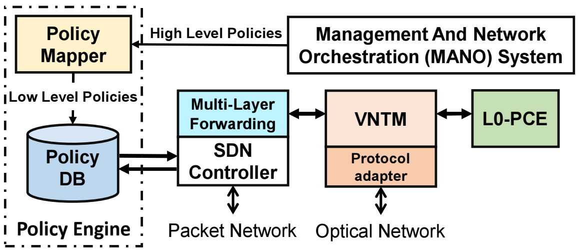

Fig. 1: Architecture Overview

service provisioning followed by description of the experimental infrastructure, scenarios and results.

Architecture description

The proposed software platform architecture, consists of four different modules as shown in Fig. 1. The SDN controller for packet networks, the Virtual Network Topology Manager (VNTM) [2] combined with the Layer-0 Path Computation Element (L0-PCE) for optical network provisioning and a policy-engine that dictates end-to-end routing and forwarding decisions across all network domains. The SDN controller has control over the packet-based devices including physical and virtual OpenFlow packet switches and the visibility of attached hosts/subnetworks. The policy-engine provides an interface to higher level management and network orchestration (MANO) entities such as applications-based network operations (ABNO)

[3] and NFV-MANO [4] to provide connectivity for their services in a simplified, technology agnostic way. The policy-engine consists of a persistent database (SQL or key-value map) which can be realized with scalable data-stores that are an integral part of the SDN controller. The policy-engine also converts high-level policies into network and technology specific constraints and forwarding rules to be stored in the policy-database. We propose a multi-layer forwarding application internal to the SDN controller. It interacts with the policy-engine to provide additional functionality to augment the standard Layer-2 forwarding as well as optical network provisioning. The multi-layer forwarding application requires interfaces to query the policy information from the policy database and to interact with the VNTM/L0-PCE to provision lightpaths in the optical network. In case of multi-domain network scenarios, the SDN controller collaborates with a network orchestrator to provision policy-based, dynamic and/or static end-to-end connectivity. The VNTM module contains the mapping of multi-layer connectivity and provides a RESTful northbound interface for the multi-layer forwarding application to setup lightpaths. It also implements a protocol adapter as a southbound interface to communicate directly with the optical devices. The L0-PCE is a stateful application, keeping the state of established lightpaths and the optical network topology information for routing new lightpath requests. The L0-PCE can be integrated with the VNTM as a single software instance to reduce communication delays of external interfaces. In the next section, we describe the detailed workflow of dynamic multi-layer service provisioning in the context of architectural blocks and their functionality.

Workflow

Fig. 2 shows the workflow for an end-to-end multi-layer service provisioning based on the proposed architecture. The multi-layer forwarding application is triggered by incoming traffic into an OpenFlow packet switch that does not match with existing flow entries. The OpenFlow *packet_in* message is received by the application and the packet headers are checked against the policy database. If the packet headers match with any policy database entry, a flow identifier (e.g. VLAN ID, VxLAN ID or MPLS label), service priority and the rate-limiting constraint is returned as part of the policy rule. The multi-layer forwarding application then computes the path based on the network topology available through the SDN controller. If an end-to-end path is found, the application acts as a traditional packet forwarding application, setting up flow forwarding rules on

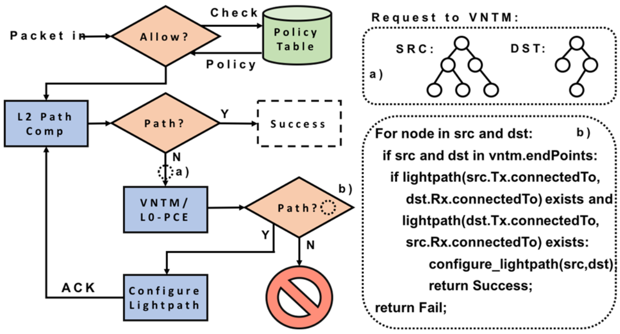

Fig. 2: End-to-End multi-layer provisioning workflow

the OpenFlow packet switches on the path. If the end-to-end path finding fails, the application considers the service priority and sends a request to the VNTM module to check if multi-layer connectivity is possible for provisioning the service. The request is a JSON structure containing two spanning trees, each with a service end-point as the root. The path computation procedure is based on the Djikstra algorithm and traverses both spanning trees and the multi-layer connectivity map of the VNTM to find the least cost end-to-end path. If the path computation algorithm suggests the creation of a new lightpath, the VNTM module configures the required lightpath with the help of the L0-PCE. The VNTM then sends back a positive acknowledgement (ACK) to the multi-layer forwarding application. This triggers the path setup procedure (i.e. setting up forwarding rules etc.), based on the new links discovered using the Link Layer Discovery Protocol (LLDP). If the VNTM sends a negative ACK, meaning the service cannot be deployed, the SDN controller discards the service request.

Experimental scenarios

In this section, we describe the experimental scenarios used for the evaluation of the proposed software platform. In scenario 1, we create a metro region data centre interconnection network consisting of 3 nodes emulated by multiple spools of optical fibre and a beam-steering fibre switch, as shown in Fig. 3. Each data centre consists of a Dell PowerEdge T630 server with 10 GbE optical connections using 1310 nm optics. In scenario 2, we use a regional section of the UK National Dark Fibre Infrastructure Service (NDFIS), as shown in Fig. 3 and the servers are connected using 10 GbE interfaces with 1550 nm based DWDM long range optics. The fibre link employed for this scenario is a 23km fibre with ~7dB loss in each direction and tuneable amplification capability available at the PoP. For this experimental scenario, we dynamically configure two cross-connections in the fibre switch to create a loopback using our software platform and also tune the EDFA for power compensation. The VNTM module configures the

8x8 fibre switch while the EDFA is dynamically configured using a REST API, facilitating correct

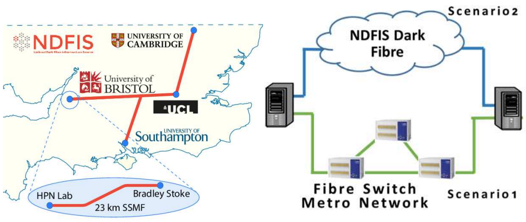

Fig. 3: Experimental scenarios

transmission of the signal. The packet switches in both network scenarios are connected in a mesh topology and deployed in the virtualised environment using OpenVSwitch (OVS) to provide the intra-DC connectivity. The hosts are implemented using Linux containers and virtual machines managed by Docker and OpenStack.

Experimental results

The proposed software platform is implemented using ONOS open source SDN controller [5] and our implementation of the VNTM module and L0-PCE. The policy engine is implemented partially as part of the MANO system which contains the policy mapper and partially as part of the SDN controller that contains the policy database. The multi-layer forwarding application is developed as a module of the ONOS controller and is modified to trigger an optical lightpath creation using VNTM/L0-PCE if the end-to-end path is not configured. We also modified ONOS to create and maintain a persistent database for the policy engine and expose a command line interface (CLI) to the policy mapper of the MANO system. Table 1 shows the structure of the initial implementation of our policy database. The MAC addresses allow end-to-end Layer-2 paths while VLANs are used to isolate multiple tenants. The priority specifies the importance of the service.

MAC Origin	MAC Dest	Priority	VLAN
00:2a:6c:d3:41:d4	cd:ac:6f:87:da:11	3	5
1b:2a:6f:3b:1e:11	96:2d:4a:c2:3f:c3	2	7
...	...	...	...

Table 1: Policies database structure

Fig. 4 shows a packet trace of the workflow described in the previous section. When a switch receives a new flow, it is sent to the controller as an OpenFlow *packet_in* message. The multi-layer forwarding application makes an HTTP POST request to the VTNM after checking against the policy database. The HTTP requests to the VNTM and the EDFA amplifier and SCPI configuration of the Polatis can also be visualized. Once the topology is updated, the packet is sent through the appropriate port (OF *packet_out*) and a flow is added to the corresponding switch. The process is repeated for the first packet in each switch for each direction except for the call to the VNTM. Once the flows are established the controller no longer requires to take any action unless the rules in the switches expire by idle timeout.

Time	Source	Destination	Protocol	Info
0.000000	137.222.204.71	137.222.204.75	OF 1.0	of_packet_in
0.175235	137.222.204.208	137.222.204.208	HTTP	POST /vntm/vlink/
0.220676	137.222.204.208	195.194.3.194	TCP	51108 > scpi-raw
0.286931	137.222.204.208	195.194.3.195	HTTP	GET /global_comm
0.982983	137.222.204.208	137.222.204.208	HTTP	Continuation or
1.406562	137.222.204.71	137.222.204.75	OF 1.0	of_packet_out
1.410255	137.222.204.208	137.222.204.221	OF 1.0	of_flow_add

Fig. 4: Packet trace of the service provisioning workflow

Fig. 5 shows the distribution of various latencies for the provisioning of end-to-end paths. In case of the metro network scenario, significant time (~400 ms) is spent for the topology update. This is caused by the delay inherent by LLDP which must be performed by the OpenVSwitch before neighbour information can be sent to the controller. In case of NFDIS scenario, the configuration of optical network, including fibre switches and EDFA amplifiers takes significant time (~850 ms) due to the multi-hop routed path between the VNTM module and the management network of the NDFIS equipment.

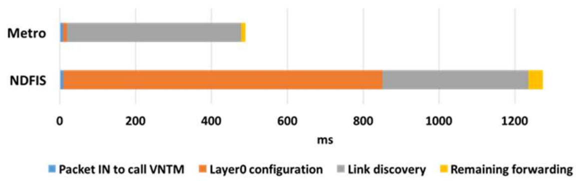

Fig. 5: Provisioning latency distribution

Conclusions

In this paper, we have proposed, implemented and successfully demonstrated a multi-layer service provisioning platform architecture incorporating a policy-based engine for end-to-end service automation over a metro and core network scenario. The experimental evaluation of the proposed platform validates the feasibility and advantages of the architecture.

Acknowledgements

This work acknowledges EP/L020009/1: Towards Ultimate Convergence of All Networks, NS/A000021/1: National Dark Fibre Infrastructure and Bristol is Open LTD.

References

[1] R. Muñoz et al., "Transport Network Orchestration for End-to-End Multilayer Provisioning Across Heterogeneous SDN/OpenFlow and GMPLS/PCE Control Domains", in JLT, April 2015

[2] "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, Sept. 2009

[3] "A PCE-Based Architecture for Application-Based Network Operations" RFC 7491, March 2015

[4] ETSI GS NFV 002 V1.2.1 "Network Functions Virtualisation (NFV); Architectural Framework" (2014-12)

[5] ONOS: http://onosproject.org/